JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 72, No. 3, MARCH 1992

Deflated Krylov Subspace Methods for Nearly Singular Linear Systems[1]

J. C. MEZA[2] AND W. W. SYMES[3]

Communicated by R. A. Tapia

Abstract. This paper concerns the use of Krylov subspace methods for the solution of nearly singular nonsymmetric linear systems. We show that the incomplete orthogonalization methods (IOM) in conjunction with certain deflation techniques of Stewart, Chan, and Saad can be used to solve large nonsymmetric linear systems which are nearly singular.

Key Words. Iterative methods, Krylov subspaces, conjugate gradient methods, nonsymmetric matrices, deflated decompositions, sparse matrices.

1. Introduction

This study concerns the use of Krylov subspace methods for the solution of nearly singular linear systems. There are many problems in numerical analysis which require the solution of nearly singular linear systems. For example, in the solution of nonlinear systems by homotopy continuation methods (Ref. 1), or in nonlinear eigenvalue problems (Ref. 2), one often has to solve nearly singular linear systems. Another example arises in constrained optimization problems (Ref. 3) where the constraints may be nearly linearly dependent. Other examples include compartmental models (Ref. 4) and decomposable Markov chains (Ref. 5). We focus on a problem from seismic processing known as the velocity inversion problem (Ref. 6).

The goal of the velocity inversion problem is to determine certain parameters describing the Earth from data taken at or near the surface. The data is known as a seismogram, and we wish to determine the sound speed structure of the Earth from these measurements. Given that we have a

[1] This work was supported by the National Science Foundation, Grants DMS-8403148 and DCR-81-16779, and by the Office of Naval Research, Contract N00014-85-K-0725.

[2] Senior Member of Technical Staff, Sandia National Laboratories, Livermore, California.

[3] Professor of Mathematical Sciences, Department of Mathematical Sciences, Rice University, Houston, Texas.

functional relation defined by $F(c) = s$, where c is the speed of sound in the Earth and s is a seismogram, the inverse problem is: given s, solve for c. Since the seismogram is usually contaminated with noise, we actually solve a nonlinear least-squares problem, min $J = \|s - F(c)\|$, for a suitable class of c.

A popular choice for the solution of nonlinear least-squares problems is the Gauss–Newton method (Ref. 7). This method computes a sequence of iterates from the formulas

$$(J(c_k)^T J(c_k))\Delta c = -J(c_k)^T F(c_k), \tag{1}$$

$$c_{k+1} = c_k + \Delta c, \tag{2}$$

for $k = 0, 1, \ldots,$ starting from an initial guess or model c_0 for the sound speed structure. In the context of the velocity inversion problem, the elements of the Jacobian matrix $J(c_k)$ are not readily available. However, it can be shown that the Jacobian matrix acting on a vector can be computed from the solution of a certain boundary-value problem. Similarly, the transpose of the Jacobian acting on a vector can be computed from the solution of a related boundary-value problem. The solutions of the two boundary-value problems are independent, which results in a linear system that is almost symmetric. The Gauss–Newton Hessian $J(c_k)^T J(c_k)$ would be symmetric, except that the solution of the two boundary-value problems which define the Jacobian matrix and its transpose can only be solved up to some nonsymmetric discretization error.

The linear systems (1) that arise in the solution of the velocity inversion problem by the Gauss–Newton method are usually very large; and since the matrix elements are not readily available, the only recourse is to use an iterative technique for the solution of the linear systems. Additionally, due to the physics of the problem, the linear systems may be ill-conditioned either by having several large singular values or by having several small singular values (Ref. 6). The small singular values arise because the initial data are band-limited. Perturbations in the model problem corresponding to frequencies outside the passband of the input data are simply not seen by the model. The large singular values are also inherent in this formulation of the problem.

In this study, we address the issues of computing the solution to the linear systems (1) by using an iterative technique which computes the solution in the space spanned by the orthogonal complement of the singular vectors corresponding to the small singular values.

Several methods have been proposed for solving nearly singular linear systems. Consider the system of linear equations

$$Ax = b, \tag{3}$$

where x and b are n-dimensional vectors and A is an $n \times n$ real matrix which has rank n. We denote the set of singular values by

$$\sigma(A)=\{\sigma_1(A),\ldots,\sigma_n(A)\}.$$

The singular values are ordered so that

$$\sigma_1\geq\sigma_2\geq\cdots\geq\sigma_n.$$

The solution to (3) can be written as

$$x=x_d+(u_n^Tb/\sigma_n)v_n, \tag{4}$$

$$x_d=\sum_{i=1}^{n-1}(u_i^Tb/\sigma_i)v_i, \tag{5}$$

where the vectors u and v are left and right singular vectors of A, respectively. The vector x_d is called the deflated solution to (3), and (4) is called the deflated decomposition. There are many definitions of the deflated solution. Chan (Ref. 8) defines deflated solutions of (3) as solutions to nearby singular but consistent systems derived from (3). The choice of the nearby system will greatly affect the deflated solution. For example, one might choose the nearest singular matrix to A in the Frobenius norm and pick the deflated solution to be the one with minimum norm. It is well known that this choice amounts to setting the smallest singular value of A equal to zero in the singular value decomposition of the matrix A.

In certain applications (Ref. 9), it is preferable to compute the decomposition (4). In other applications, the deflated solution is the only solution of interest. Notice that, if the eigenvector u_n were known, then both (4) and (5) could be computed by first computing x and then orthogonalizing against u_n. However, even if the eigenvector u_n were known, this approach is not advisable because it usually results in a poor approximation to x_d due to roundoff errors. In particular, if the component of the solution in the direction of the null eigenvector is large, then errors in that component tend to dominate the solution in the other directions.

Stewart (Ref. 10) suggested a method for computing the deflated solution of (3) by an implicit method. This method uses orthogonal projections constructed from approximations to the singular vectors of the matrix A corresponding to the smallest singular value. The disadvantage of this method is that it requires a direct method for the solution of (3). Chan and Saad (Ref. 11) proposed a deflated Lanczos method for symmetric, positive-definite linear systems which only requires the matrix-vector product Ax. The goal of this paper is to study methods for computing the deflated solution for large sparse nonsymmetric linear systems which are nearly singular.

The basic idea is to use one of the Krylov subspace methods proposed by Saad (Ref. 12) for solving nonsymmetric linear systems. These methods compute an approximation to the solution by generating iterates which lie in a certain Krylov subspace. The approximations to the solution are then computed by solving a linear system described by a small upper Hessenberg matrix H, produced by Arnoldi's method (Ref. 13).

Arnoldi's method may be thought of as a Galerkin process for approximating the eigenvalues of A by the eigenvalues of the upper Hessenberg matrix H. Like the Lanczos method, the approximations to the eigenvalues tend to be best at the extremes of the spectrum, so that the matrix H may be expected to also be nearly singular. Therefore, one disadvantage to using the methods proposed by Saad is that they could require the solution of a nearly singular linear system. Unlike the original system (3), however, we need only to be able to solve linear systems involving the much smaller matrix H. We can then use the deflation techniques suggested by Stewart and Chan. In addition, we propose another method based on solving a truncated least-squares problem, analogous to the GMRES method suggested by Saad and Schultz (Ref. 14).

2. Krylov Subspace Methods

Saad (Ref. 12) proposed a class of methods for solving large sparse nonsymmetric linear systems based on the Arnoldi process (Ref. 13) for computing the eigenvalues of a matrix. Arnoldi's method is a generalization of the Lanczos method (Ref. 15) for nonsymmetric matrices, and when it is applied to a symmetric matrix it reduces to the Lanczos method. Like the Lanczos method, Arnoldi's method is best viewed as an iterative method for approximating the eigenvalues of large sparse matrices. In essence, Arnoldi's method is just the Gram-Schmidt method for computing an orthonormal basis for the Krylov subspace $\kappa_m(w_1, A) \equiv \text{span}\{w_1, Aw_1, \ldots, A^{m-1}w_1\}$. The method can be described as follows.

```
Algorithm 2.1. Arnoldi's Method.
Step 1. Choose $w_1$ such that $\|w_1\| = 1$.
Step 2. For $j = 1, 2, \ldots$, compute
    $h_{ij} = (Aw_j, w_i), \qquad i = 1, 2, \ldots, j,$
    $\hat{w}_{j+1} = Aw_j - \sum_{i=1}^{j} h_{ij} w_i,$
    $h_{j+1,j} = \|\hat{w}_{j+1}\|,$
    $w_{j+1} = \hat{w}_{j+1} / h_{j+1,j}.$
```

If we let

$$W_m = [w_1, w_2, \ldots, w_m],$$

then it is easy to show that

$$H_m = W_m^T A W_m, \tag{6}$$

where the entries of the upper Hessenberg matrix H_m are the scalars h_{ij} produced by Arnoldi's method after m steps.

As Saad (Ref. 12) has shown, Arnoldi's method may be used as a basis for a class of Krylov subspace methods for solving large sparse nonsymmetric linear systems. By a Krylov subspace method, we mean any method that approximates the solution to the linear system (3) by generating iterates of the form

$$x_m = x_0 + z_m,$$

where x_0 is an initial guess, $z_m \in \kappa_m(r_0, A)$, and $r_0 = b - Ax_0$. If we carry out m steps of Arnoldi's method starting with $w_1 = r_0/\|r_0\|$, and if we impose the Galerkin condition that the residuals at each iteration be orthogonal to $\kappa_m(r_0, A)$, then this yields

$$W_m^T A W_m y_m - W_m^T r_0 = 0.$$

Using the relation (6) yields

$$x_m = x_0 + W_m y_m,$$

where y_m solves the system

$$H_m y_m = \|r_0\| e_1, \tag{7}$$

and

$$e_1 = (1, 0, \ldots, 0)^T.$$

This defines the full orthogonalization method (Ref. 12) for solving (3).

Algorithm 2.2. Full Orthogonalization Method.

Step 1. Choose x_0, and compute $r_0 = b - Ax_0$. Set $w_1 = r_0/\|r_0\|$.

Step 2. For $j = 1, 2, \ldots, k$, compute

$$h_{ij} = (Aw_j, w_i), \qquad i = 1, 2, \ldots, j,$$

$$\hat{w}_{j+1} = Aw_j - \sum_{i=1}^{j} h_{ij} w_i,$$

$$h_{j+1,j} = \|\hat{w}_{j+1}\|,$$

$$w_{j+1} = \hat{w}_{j+1}/h_{j+1,j}.$$

Step 3. Solve $H_k y_k = \|r_0\| e_1$.

Step 4. Set $x_k = x_0 + W_k y_k$.

In practice, the number of iterations is chosen so that the approximate solution x_k is sufficiently accurate. Usually, this is measured by requiring that the initial residual be decreased by a user-specified amount. Fortunately, the residual at any iteration may be computed without actually computing the solution to (3) through the relation (Ref. 12)

$$\|b - Ax_k\| = h_{k+1,k} |e_k^T y_k|. \tag{8}$$

Although the computation of the residual by (8) requires solving (7) for y_k, there are ways to circumvent this computation by carrying an LU or QR factorization of H throughout the Arnoldi process. If, after k iterations, the approximate solution has not converged, then it is possible to restart the algorithm using the current estimate of x as the new initial guess. This method is denoted by FOM(k) or the restarted FOM.

It is well known that the Arnoldi process may be viewed as a Galerkin process for estimating the eigenvalues of a matrix. In particular, if we apply Algorithm 2.2 to a linear system that is nearly singular, then the upper Hessenberg matrix H_k, which is generated after k steps of the Arnoldi process, will probably have a small eigenvalue. Therefore, if we solve (7) for y_k in the straightforward way, our computed solution will be inaccurate for the reasons indicated in Section 1.

Fortunately, computing the deflated solution of (7) is easier than computing the deflated solution of (3). Since the matrix H_k has dimension $k \ll n$, the solution of (7) is at least computationally easier. Moreover, the matrix elements of H_k are on hand, whereas the matrix elements of A are not available. The next section describes some of the deflation techniques which can be used to compute the deflated solution of (7) in a stable manner.

3. Deflation Methods

This section describes three methods for computing the deflated solution to a nearly singular linear system. In particular, we consider the linear system

$$H_k y = f, \tag{9}$$

where y and f are k-dimensional vectors and H_k is the upper Hessenberg matrix generated after k steps of Arnoldi's method. Further, we assume that H_k is nearly singular with a rank deficiency of at most one. The extensions to null spaces of dimensions greater than one will be treated in a later section.

The first method, which uses orthogonal projections, was proposed by Stewart (Ref. 10). The second method is a generalization of a technique suggested by Chan and Saad (Ref. 11) for symmetric positive-definite systems. In essence, this method uses a QR iteration to decouple the linear system into a well-conditioned problem plus a component corresponding to the small eigenvalue. The third method computes the deflated solution by solving a truncated least-squares problem.

3.1. Deflation by Orthogonal Projection. Stewart (Ref. 10) proposed an implicit method for computing the deflated solution to a nearly singular linear system. His algorithm consists of constructing two orthogonal projectors defined by approximations to the singular vectors corresponding to the small singular value.

Consider the right and left singular vectors, respectively v and u, corresponding to the smallest singular value of H_k. Define the orthogonal projectors

$$P_u = I - uu^T \quad \text{and} \quad P_v = I - vv^T.$$

The projector $P_u(P_v)$ is merely the orthogonal projector onto the orthogonal complement of the space spanned by $u(v)$. It is easy to show that the deflated solution to (9) is the unique vector satisfying the relations

$$P_u H_k P_v y_d = P_u f, \tag{10}$$

$$P_v y_d = y_d. \tag{11}$$

Stewart suggests using the following algorithm based on iterative refinement to solve for y_d.

Algorithm 3.1. Deflation by Orthogonal Projection.

```
Step 1. Set $y = 0$.
Step 2. For $k = 1, 2, \ldots,$
            solve $H_k d = P_u(f - H_k y)$;
            set $y = y + P_v d$.
Step 3. Set $y_d = y$.
```

Since the singular vectors are not known, Stewart suggests approximating u and v by a variant of the inverse power method. In the case of interest, where the small singular value is isolated, the inverse power method is known to converge rapidly. Stewart also gives conditions, which depend on the accuracy attained in the approximation to the vectors u and v, under which Algorithm 3.1 will converge to the deflated solution of (9).

3.2. Deflation by QR Iteration. The second method for computing the deflated solution is a generalization of a method proposed by Chan and Saad (Ref. 11) for symmetric, positive-definite systems. To motivate the discussion, assume that we have the pair (λ_k, u_k) of the unreduced upper Hessenberg matrix H_k. It is well known that one step of the shifted QR iteration method with a shift of λ_k reduces the matrix H_k so that the eigenvalue λ_k appears on the diagonal and the corresponding subdiagonal element is zero. The linear system (9) can then be decoupled so that it is easily solved for the deflated solution.

In particular, assume that we compute the matrix $H^{(1)}$ by the following two-step procedure:

Step 1. Compute the QR factorization of $H_k - \lambda_k I$.

Step 2. Form $H^{(1)} = RQ + \lambda_k I$.

Then, the matrix $H^{(1)}$ takes on the form

$$H^{(1)} = \begin{array}{c} \\ k-1 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} k-1 & 1 \end{array} \\ \begin{bmatrix} \hat{H} & \hat{h} \\ 0 & \lambda_k \end{bmatrix} \end{array},$$

where $\hat{H}$ is an upper Hessenberg matrix of order $k-1$. It is easy to show that

$$H^{(1)} = Q^T H_k Q,$$

so that we can transform (9) into

$$H^{(1)} z = \hat{f}, \tag{12}$$

where

$$y = Qz \quad \text{and} \quad \hat{f} = Q^T f.$$

If we partition the vectors z and $\hat{f}$ so that

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \qquad \hat{f} = \begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \end{bmatrix},$$

then the deflated solution to (12) is

$$z_d = \begin{bmatrix} \hat{H}^{-1} \hat{f}_1 \\ 0 \end{bmatrix}. \tag{13}$$

The deflated solution to (9) can now be computed from $y_d = Qz_d$.

There are two remarks in order. The first is that in general we do not know the eigenvalue λ_k; however, as in the previous section, we may compute an approximation to λ_k by using the inverse power method. The second remark is that, even if we had an exact value for λ_k, in practice the matrix H_k will not be reduced in one QR iteration step. Wilkinson (Ref. 16) suggests iterating with the shifted QR method until the element on the subdiagonal has converged to zero. We use the test

$$\|H^{(1)}_{k,k-1}/H^{(1)}_{k,k}\| \leq \epsilon \tag{14}$$

to check for convergence. Two or three iterations are usually sufficient to satisfy (14). This leads to the following algorithm.

Algorithm 3.2. Deflation by QR Iteration.

```
Step 1.  Compute $\lambda_k$ from $H_k$ via inverse iteration.
Step 2.  $H^{(0)} = H_k$.
Step 3.  For $j = 0, 1, \ldots,$ until convergence:
             compute the QR factorization of $H^{(j)} - \lambda_k I$;
             compute $H^{(j+1)} = RQ + \lambda_k I$.
Step 4.  Set $z_d = [\hat{H}^{-1}\hat{f}_1, 0]^T$.
Step 5.  Set $y_d = Qz_d$.
```

If the dimension of the null space is greater than 1, then the issues become more complicated. There are two cases to consider: a small eigenvalue of multiplicity greater than 1 and the case of several distinct small eigenvalues. The case of a small real eigenvalue with a multiplicity greater than 1 is easily handled, since the shifted QR method will still converge. The case of several distinct small eigenvalues is harder to address since the QR method does not guarantee that the converged eigenvalues will appear in any particular order on the diagonal of $H^{(1)}$. A related issue is that of complex eigenvalues. Since the inverse power method converges to the eigenvalue of smallest modulus, we must be careful in choosing the shift. A good choice would be to use Francis' implicit double shift QR method and with a shift as described by Wilkinson (Ref. 16).

3.3. Deflation by Truncated Least Squares. The last method that we discuss is derived from a technique for solving rank-deficient least-squares problems (Ref. 17, pp. 162–167). The general idea in these problems is to compute a QR factorization of the nearly singular matrix H_m such that the elements on the diagonal of the matrix R display the rank deficiency. The solution to the linear system (9) is then computed by setting the small diagonal elements of the matrix R equal to zero and solving a truncated least-squares problem.

Assume that the matrix H_k has exactly one zero eigenvalue and consider its QR factorization,

$$H_k\Pi = QR,$$

where Π is a permutation matrix chosen so that the matrix R has the form

$$R = \begin{array}{c c} & \begin{array}{cc} k-1 & 1 \end{array} \\ \begin{array}{c} k-1 \\ 1 \end{array} & \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix} \end{array},$$

where R_{11} is upper triangular.

In this case, the least-squares solution to (9) can be easily computed. In fact,

$$\|H_k y - f\|^2 = \|R_{11}z_1 - (\hat{f}_1 - R_{12}z_2)\|^2 + \|\hat{f}_2\|^2,$$

where

$$\Pi^T y = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \tag{15}$$

$$Q^T f = \begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \end{bmatrix}. \tag{16}$$

If we set $z_2 = 0$, then the solution is given by

$$\hat{y} = \Pi \begin{bmatrix} R_{11}^{-1}\hat{f}_1 \\ 0 \end{bmatrix}. \tag{17}$$

The vector $\hat{y}$ is called the basic solution. In general, it is not the least-norm least-squares solution unless $R_{12} = 0$.

In practice, the matrix H_k will never have an eigenvalue exactly equal to zero. The question is then to determine the rank of the matrix H_k from the elements of R. One popular choice is the method of column pivoting implemented in LINPACK (Ref. 18). This method is usually reliable in detecting the rank deficiency of a matrix, although there are counterexamples where the method may fail. Chan (Ref. 19) suggested another method for producing a QR factorization which guarantees a small R_{22} element. The essential ideas can be explained in the case of a rank-one deficient matrix. First, we need the following theorem proved by Chan (Ref. 19).

Theorem 3.1. Suppose that $x \in \mathbb{R}^n$ with $\|x\| = 1$ such that $\|Ax\| = \epsilon$. Let Π be a permutation such that, if $\Pi^T x = y$, then $|y_n| = \|y\|_\infty$. Then, if $A\Pi = QR$ is the QR factorization of $A\Pi$, we have

$$|r_{nn}| \leq \sqrt{n}\,\epsilon.$$

The usefulness of this theorem is apparent if we consider the right singular vector v of the matrix H_k corresponding to the smallest singular value σ_k. Then, we have

$$\|v\|=1 \quad \text{and} \quad \|H_k v\|=\sigma_k.$$

If we define the permutation Π by

$$|(\Pi^T v)_k|=\|v\|_\infty,$$

then $H_k\Pi=QR$ has a pivot r_{mm} at least as small as $\sqrt{m}\sigma_m$ in absolute value. As in Stewart's method, all that is required is an approximation to v, which may be computed by the inverse power method. This suggests the following algorithm for solving (9).

Algorithm 3.3. Deflation by Truncated Least Squares.

Step 1. Compute the QR factorization of H_k.

Step 2. Compute v and σ_k from H_k via inverse power method.

Step 3. Compute Π so that $|(\Pi^T v)_k|=\|v\|_\infty$.

Step 4. Compute the QR factorization of $H_k\Pi$.

Step 5. Compute $\hat{y}=\Pi[R_{11}^{-1}\hat{f}_1, 0]^T$.

This method has the obvious advantage of being immediately applicable to null spaces with a dimension greater than 1. However, the question of rank deficiency is still a hard problem and the user must be able to supply a tolerance which specifies the amount of ill-conditioning allowed.

4. Numerical Results

This section presents several numerical experiments comparing the various methods described in Section 3.

Recall that the linear system of interest is

$$Ax=b,$$

where x and b are n-dimensional vectors and A is an $n\times n$ real matrix which is nearly singular. All of the numerical results presented here are for linear systems of order 10. The results are similar for larger systems. The numerical experiments were run on a Sun 3/160 computer, using single precision arithmetic (machine epsilon $\approx 10^{-7}$). The method was said to converge whenever

$$\|r_k\|\le 10^{-6}\cdot\|r_0\|.$$

The basic method used was the restarted FOM(k). To avoid the issue of deciding when the eigenvalues of the upper Hessenberg matrix H are good approximations to the eigenvalues of the matrix A, we set $k=n$.

Problem 1. The first test case was a small perturbation to a symmetric, positive-definite matrix. Define

$$A(\epsilon)=D+\epsilon E.$$

The matrix D is defined by

$$D=\operatorname{diag}(10^{-I}, 2, 3, \ldots, n),$$

and I varies from 1 to 7. The nonsymmetric perturbations are generated using the random number generator URAND from IMSL. The matrices E are computed by generating uniform random numbers between $[-0.5, +0.5]$, and normalizing so that $\|E\|_2=1$. The amount of nonsymmetry can then be adjusted by varying ϵ. We set $\epsilon=10^{-3}$.

Problem 2. The second set of test cases was picked from a study done by Chan and Saad (Ref. 11). In this set of problems, we set

$$A=(I-uu^T)D(I-vv^T),$$

where u and v were chosen randomly with the constraint that they have norm one. The matrix D was chosen as in Problem 1.

Problem 3. The purpose of the third problem is to simulate a typical linear system arising in the velocity inversion problem. These problems usually have one or more small singular values and one or more large singular values with the rest of the spectrum fairly well-conditioned. In this problem, we set the matrix

$$D=\operatorname{diag}(10^{-J}, 1, \ldots, 3, 3000),$$

with the values of d_2 through d_{n-1} varying uniformly between 1 and 3. The matrix A is then computed as in Problem 1. This example generates a well-conditioned problem if the small and large eigenvalues are excluded, which is typical of some of the velocity inversion problems.

The four methods described so far are:

Method 1. Full orthogonalization method (FOM).

Method 2. Full orthogonalization method, with Stewart's orthogonal projection deflation (FOMOP).

Method 3. Full orthogonalization method with the truncated least-squares deflation (FOMLSQ).

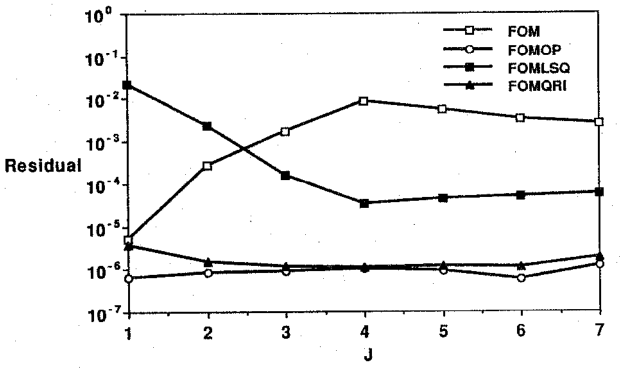

Fig. 1. Norm of the residual, Problem 1.

Method 4. Full orthogonalization method, with the QR iteration deflation (FOMQRI).

In addition, we also tested an algorithm based on computing the solution to the linear system by the full orthogonalization method and then computing the deflated solution by orthogonalizing against the null vector. This method will be referred to as FOM.

Figures 1-3 plot $\log\|r\|$ versus J. Since we are interested in the deflated solution, we have chosen to plot the norm of the residual corresponding to the deflated solution. As the value of J increases, the linear systems become more ill-conditioned, with the smallest singular value of each matrix approximately equal to 10^{-J}. The effect of the near singularity of the linear systems on Method 1 is apparent as the plots show the norm of the residual increasing as the linear system becomes more ill-conditioned. This is to be

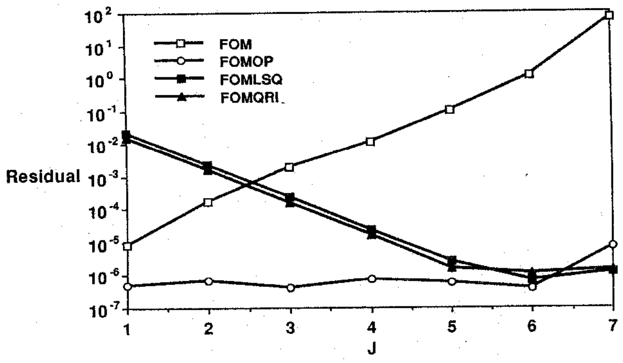

Fig. 2. Norm of the residual, Problem 2.

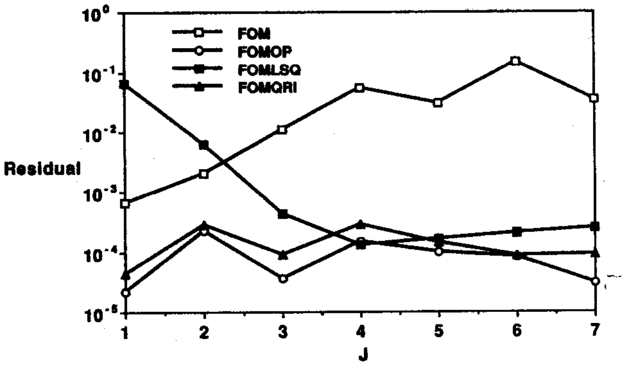

Fig. 3. Norm of the residual, Problem 3.

expected, as the error in the component corresponding to the smallest singular values tends to contaminate the rest of the solution. Method 2, FOMOP, is clearly the best method in terms of producing the smallest residuals. Method 3, FOMLSQ, was disappointing in that the norm of the residual was consistently worse than the other methods. Method 4, FOMQRI, was inconsistent in this set of problems. In Problems 1 and 3, FOMQRI was almost as good as FOMOP. However in Problem 2 the method performed much worse.

The question of extending these methods to null spaces of higher dimensions is also of interest. In this respect, FOMLSQ can be easily extended while the other methods would require some extra work. In fact, the ease with which FOMLSQ can be extended to solving systems which have several small singular values is perhaps the only redeeming factor of this method. Figures 4–6 illustrate the effect of solving the same systems of

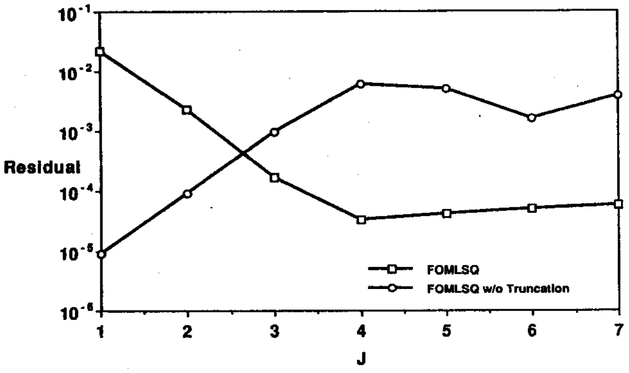

Fig. 4. Effect of truncation on FOMLSQ, Problem 1.

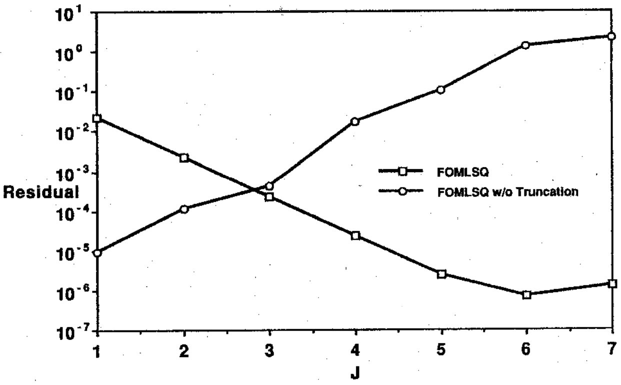

Fig. 5. Effect of truncation on FOMLSQ, Problem 2.

Problems 1-3 using both a truncated least-squares approach and simply solving the system using a full least-square problem. It can be seen that the two curves usually intersect between $J = 2$ and $J = 3$, which corresponds to $\kappa(A) \approx 1000$. This implies that using FOMLSQ with a properly chosen tolerance would perform better than always solving the truncated least-squares problem. Fortunately, this is the case of interest. This approach would still not be as good as using FOMOP, but it has the advantage that it is easier to extend to null spaces with dimension greater than 1, whereas extending FOMOP to handle several small singular values would require extra work.

Further experiments remain to be done. It is not clear which deflated solution is best in terms of the norm of the error. In fact, there are many

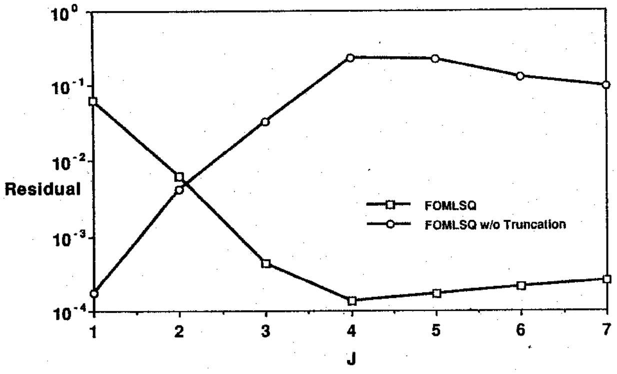

Fig. 6. Effect of truncation on FOMLSQ, Problem 3.

definitions of the deflated solution, and the choice of the deflated solution greatly alters the results. That is one reason we have chosen to work with the norm of the residual. Another possibility lies in solving the upper Hessenberg system by using the singular-value decomposition. Although this may seem like too much work at first glance, the problems we are dealing with have the property that one matrix-vector multiplication is very expensive. In this context, computing the SVD for a small upper Hessenberg matrix would be insignificant. This approach also has the added attraction that, like the FOMLSQ method, it is easily extended to null spaces of dimension greater than 1. This research will be the subject of a later report.

References

1. ALLGOWER, E., and GEORG, K., *Simplicial and Continuation Methods for Approximating Fixed Points and Solutions to Systems of Equations*, SIAM Review, Vol. 22, pp. 28-85, 1980.
2. CHAN, T. F., *Newton-Like Pseudo-Arclength Methods for Computing Simple Turning Points*, SIAM Journal on Scientific and Statistical Computing, Vol. 5, pp. 135-148, 1984.
3. GILL, P. E., MURRAY, W., and WRIGHT, M., *Practical Optimization*, Academic Press, New York, New York, 1979.
4. FUNDERLIC, R. E., and MANKIN, J. B., *Solution of Homogeneous Systems of Linear Equations Arising from Compartmental Models*, SIAM Journal on Scientific and Statistical Computing, Vol. 2, pp. 375-383, 1981.
5. STEWART, G. W., *Computable Error Bounds for Aggregated Markov Chains*, Computer Science Technical Report 901, University of Maryland, 1980.
6. SYMES, W. W., *Stability Properties for the Velocity Inversion Problem*, Paper Presented at the SEG/SIAM/SPE Symposium, Houston, Texas, 1985.
7. DENNIS, J. E., and SCHNABEL, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.
8. CHAN, T. F., *Deflated Decomposition of Solutions of Nearly Singular Systems*, SIAM Journal on Numerical Analysis, Vol. 21, pp. 738-754, 1984.
9. CHAN, T. F., *Deflation Techniques and Block-Elimination Algorithms for Solving Bordered Singular Systems*, SIAM Journal on Scientific and Statistical Computing, Vol. 5, pp. 121-134, 1984.
10. STEWART, G. W., *On the Implicit Deflation of Nearly Singular Systems of Linear Equations*, SIAM Journal on Numerical Analysis, Vol. 2, pp. 136-140, 1981.
11. CHAN, T. F., and SAAD, Y., *Deflated Lanczos Procedures for Solving Nearly Singular Systems*, Computer Science Report YALEU/DCS/RR-403, Yale University, 1985.
12. SAAD, Y., *Krylov Subspace Methods for Solving Large Unsymmetric Linear Systems*, Mathematics of Computation, Vol. 37, pp. 105-126, 1981.

13. ARNOLDI, W. E., *The Principle of Minimized Iteration in the Solution of the Matrix Eigenvalue Problem*, Quarterly of Applied Mathematics, Vol. 9, pp. 17–29, 1951.
14. SAAD, Y., and SCHULTZ, M. H., *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*, SIAM Journal on Scientific and Statistical Computing, Vol. 7, pp. 856-869, 1986.
15. LANCZOS, C., *An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators*, Journal of Research of the National Bureau of Standards, Vol. 45, pp. 255–282, 1950.
16. WILKINSON, J. H., *The Algebraic Eigenvalue Problem*, Oxford University Press, London, England, 1965.
17. GOLUB, G., and VAN LOAN, C., *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 1983.
18. DONGARRA, J. J., BUNCH, J. R., MOLER, C. B., and STEWART, G. W., *LINPACK Users' Guide*, SIAM Press, Philadelphia, Pennsylvania, 1984.
19. CHAN, T. F., *Rank Revealing QR Factorizations*, Computer Science Report YALEU/DCS/RR-398, Yale University, 1985.